Title: Clock Offset Estimation with Bias Correction

Field of the Invention

The present invention relates to synchronization of clocks. More specifically, the present invention relates to estimation and correction of clock offset in distributed resources interconnected by a network, such as the Internet.

Background of the Invention

Accurate and reliable time information is necessary for many systems and applications involving distributed resources, including networked systems and processes typified by the Internet. In such systems, different functional elements are required to have their clocks synchronized. Clock synchronization involves two aspects: frequency synchronization and time synchronization. The former means that element clocks run at the same frequency, and the latter means that elements agree at a particular epoch with respect to the coordinated universal time (UTC), *i.e.*, there is no *offset* between element clocks. For many purposes, it is appropriate to focus on estimating clock offset and to assume higher order effects, such as the frequency offset, can be ignored or provided for separately.

Clock synchronization issues have been extensively addressed in the literature. See, for example, D. Mills, Internet time synchronization: the Network Time Protocol, *IEEE Trans. Communications*, Vol. 39, No.10, October 1991; D. Mitra, Network synchronization: analysis of a hybrid of master-slave and mutual synchronization, *IEEE Trans. Communications*, COM-28, 8 (August 1980), pp.1245-1259; and N.W. Rickert, Non Byzantine clock synchronization – a programming experiment, *ACM Operating Systems Review* 22,1 (January 1988), pp.73-78.

A well-known clock synchronization protocol that has been successfully deployed in the Internet is the Network Time Protocol (NTP), described, for example, in D. Mills, *Network Time Protocol (version 3) Specification, Implementation and Analysis RFC* 1305, March 1992. One of the most important network clock synchronization issues addressed by NTP is how to use the collected data to estimate the clock offset between a pair of network elements.

In V. Paxson, On Calibrating Measurements of Packet Transit Times, LBNL-41535, ftp://ftp.ee.lbl.gov/papers/vp-clocks-sigmetrics98.ps.gz, March, 1998 (and in a

shortened paper with the same title published in *Proc. ACM Sigmetrics98*, June 22-26, 1998), the author proposed a new algorithm for clock offset estimation. For easy reference, this algorithm will be referred to as the Separate Direction Estimation Algorithm (SDEA). While SDEA can provide improved performance relative to the NTP algorithm, SDEA nevertheless suffers from significant limitations, especially in applying SDEA to contexts in which loading is different for each of the directional links between pairs of network elements.

Summary of the Invention

Limitations of the prior art are overcome and a technical advance is made in accordance with the present invention, illustrative embodiments of which are presented in the following detailed description. In particular, limitations of the NTP algorithm and SDEA are overcome and SDEA techniques are extended and improved in accordance with present inventive teachings.

In addition to estimating the clock offset, present inventive techniques also estimate the bias of the estimation and attempt to correct any such bias. As a result, present inventive algorithms show significant improvement in terms of convergence speed and accuracy. Illustrative embodiments of present inventive algorithms will be referred to as Separate Direction Estimation Algorithms with Bias Correction (SDEABC).

In accordance with one aspect of present inventive methods, messages are exchanged (bi-directionally) between pairs of network elements, such messages including timestamps indicative of sending and receiving times noted at each stop. Because variable components of delay for each message direction need not be characterized by identical probability distribution functions, *e.g.*, when links in each direction are differently loaded, undesired bias of estimates for clock offset can emerge. Illustrative embodiments of the present invention avoid errors in estimates of variable delay minimums by separately determining these minimums, *e.g.*, by advantageously employing separate empirical probability distributions for each direction.

Brief Description of the Drawing

FIG. 1 illustrates time relations in sending messages between network elements.

FIG. 2 shows an illustrative network embodiment of the present invention having a single server node for interacting with a plurality of other nodes to effect clock correction at such other nodes.

FIG. 3 shows an illustrative network embodiment of the present invention having a hierarchy of server nodes with lowest order server nodes for interacting with other network nodes to effect clock correction at such other nodes.

Detailed Description

Separate Direction Estimation Algorithm

In network clock synchronization, each of a pair of network elements exchanges data packets (*timing messages*) with the other of the pair. One of such a pair is referred to as a *sender*, and the other as a *receiver* with respect to a particular packet. Based on time stamps contained in these timing messages, the clock offset between the sender and the receiver is estimated.

More specifically, timing messages are sent between network elements as shown in FIG. 1, where activities at a sender 100 and a receiver 110 are shown for a particular round of message exchanges. At the *i*-th round of message exchange, the *i*-th message includes a time stamp T_i^0 indicating current time as known at the sender 100 when the message is sent to the receiver 110. Immediately upon reception of this message, receiver 110 puts a time stamp T_i^1 in the received message. The receiver then puts another time stamp T_i^2 on the message immediately before sending the message back to the sender. When sender 100 receives the message, it records the receiving time T_i^3. Using $T_i^o, T_i^1, T_i^2, T_i^3$, for i = 1,2, ..., the sender computes an estimate of the clock offset as between sender 100 and receiver 110.

Let θ be the receiver's clock offset from the sender. That is, if at a given instant the receiver's clock shows time t_s and the sender's clock shows time t_c, then $\theta \triangleq t_s - t_c$. It proves convenient to denote the fixed delay from the sender to the receiver (called *upward direction*) by d^u and the fixed delay from the receiver to the sender (called *downward direction*) by d^d. In this representation, the fixed delay includes all non-variable components of the delay such as transmission delay and propagation delay.

Let e_i^u denote the variable component of the delay of the i-th message from the sender to the receiver and e_i^d denote the variable component of delay from the receiver to the sender. The variable delay component is the part of the delay that would not occur under ideal conditions, and includes such delays as packet queuing delay and delay due to the unavailability of shared resources (*e.g.*, CPU or bandwidth).

We then have the following equations

$$T_i^1 - T_i^0 = d^u + \theta + e_i^u \quad (1)$$

$$T_i^3 - T_i^2 = d^d - \theta + e_i^d \quad (2)$$

For ease of further description of present inventive algorithms, it proves convenient to define $X_i \triangleq T_i^1 - T_i^0$ and $Y_i \triangleq T_i^3 - T_i^2$, where $i = 1, ..., n$ after the exchange of n messages.

The above-cited SDEA keeps the smallest values of X_i and Y_i, *viz.*, it keeps variables

$$U_n \triangleq \min_{i=1,..,n}\{X_i\},$$

and

$$V_n \triangleq \min_{i=1,..,n}\{Y_i\}.$$

The estimate for θ using n samples is then computed as

$$\hat{\theta}_n = (U_n - V_n)/2. \quad (3)$$

Separate Direction Estimation Algorithm with Bias Correction (SDEABC)

From (3), with $E[\cdot]$ as the expected value operator we can see that, assuming $d^u = d^d$,

$$E[\hat{\theta}_n] = E[U_n - V_n]/2$$

$$= \theta + (E[\min_{i=1,...,n}\{e_i^u\}] - E[\min_{i=1,...,n}\{e_i^d\}])/2. \quad (4)$$

Therefore, the SDEA is asymptotically unbiased if the pdf of both $e_i^u \geq 0$ and $e_i^d \geq 0$ is positive near 0. However, with a finite number of samples, the estimator is biased if $E[\min_{i=1,..,n}\{e_i^u\}] \neq E[\min_{i=1,..,n}\{e_i^d\}]$, a not unlikely event because of differences between uplink

and downlink traffic loading. As will be seen, illustrative embodiments of the present invention reduce such errors in estimates due to bias.

The empirical distribution of a random variable $R \geq 0$ is illustratively constructed in a manner now to be described. Suppose R is independently sampled n times with the result being $r_1, \ldots, r_n$. We rearrange the n samples so that $r_{1:n} < r_{2:n} \leq \ldots \leq r_{n:n}$, where each $r_{i:n}$ is from the original sample set. Further, let $r_{o:n} = 0$ and $r_{n+1:n} = \infty$. The empirical distribution of R is then

$$F_n(x) = \sum_{i=1}^{n+1} (\frac{i-1}{n}) I(r_{i-1:n} \leq x < r_{i:n}), \qquad (5)$$

where $I(\cdot)$ is an indicator function, *i.e.*, $I(\cdot) = 1$ when the argument of I is satisfied, and is 0 otherwise. Equivalently, the complementary distribution function is

$$\overline{F}_n(x) = \sum_{i=1}^{n+1} (\frac{n-i+1}{n}) I(r_{i-1:n} \leq x < r_{i:n}). \qquad (6)$$

Note that since the intervals $[r_{i-1:n}, r_{i:n})$ are non-overlapping,

$$(1-F_n(x))^n = \sum_{i=1}^{n+1} (\frac{n-i+1}{n})^n I(r_{i-1,n} \leq x < r_{i,n}) \qquad (7)$$

Suppose that independent, identically distributed (i.i.d.) random variables R_i, $i=1,2,\ldots,n$ have a distribution function $F(x)$. Then, $\min_{i=1}^{n}\{R_i\}$ has the complementary distribution functions $(1-F(x))^n$. Define

$$\gamma_n^R \triangleq E[\min_{i=1}^{n}\{R_i\}],$$

By using the empirical distribution function $F_n(x)$ to replace $F(x)$, an estimate of γ_n^R, $\hat{\gamma}_n^R$, is obtained as

$$\hat{\gamma}_n^R = \int_0^{\infty} (1-F_n(x))^n dx$$

$$\hat{\gamma}_n^R = \sum_{i=1}^{n+1} (\frac{n-i+1}{n})^n * (r_{i:n} - r_{i-1:n}). \qquad (8)$$

As shown in (4), the bias of the SDEA estimator is

$$b_n \triangleq E[\hat{\theta}_n] - \theta,$$

and an estimate of this bias, $\hat{b}_n$, is given by

$$\hat{b}_n = ((\hat{\gamma}_n^X - \hat{\gamma}_n^Y)/2) - \hat{\theta}_n . \qquad (9)$$

Based on the foregoing, it can readily be seen that a more accurate determination can be made of clock offset by correcting for bias of estimates made using SDEA. That is, by determining analytically what the bias in an estimate using SDEA is in a particular environment, correcting by an amount equal to the bias achieves a more accurate estimate of clock offset. This more accurately estimated correction is then advantageously applied to the out-of-synchronization clock. Moreover, the expected value of the bias provides a monitor for the analytical process; variations of this expected value over time can signal conditions in a network that may indicate greater or lesser confidence in clock offset estimates.

In accordance with an approach that is preferred for some applications, a method again proceeds from evaluation of $\hat{\theta}_n = (U_n - V_n)/2$, as in Eq. (3). Since

$$U_n \triangleq \min_{i=1,..,n}\{X_i\},$$

and

$$V_n \triangleq \min_{i=1,..,n}\{Y_i\},$$

forming the expected value of each side of Eq. (3) and employing the notation from Eq. (8) for each of random variables U_n and V_n yields

$$\begin{aligned} E[\hat{\theta}_n] &= E[(U_n - V_n)]/2 \\ &= \left[\frac{E[U_n]}{2}\right] - \left[\frac{E[V_n]}{2}\right] \\ &= \left[\frac{\gamma_n^X}{2}\right] - \left[\frac{\gamma_n^Y}{2}\right] = \frac{1}{2}(\gamma_n^X - \gamma_n^Y) \end{aligned} \qquad (10)$$

We may take

$$b_n \triangleq E[\hat{\theta}_n] - \theta,$$

but in forming an estimate of b_n, the bias in the estimate of θ after n samples, we employ

$$\hat{b}_n = \frac{1}{2}(\hat{\gamma}_n^X - \hat{\gamma}_n^Y) - \frac{1}{2}(\min\{X_i\} - \min\{Y_i\}).$$

Then, in accordance with the present approach, a bias-corrected (BC) estimate (SDEABC) for θ is given by

$$\hat{\theta}_n^{BC} = \hat{\theta}_n - \hat{b}_n$$

$$= (\min\{X_i\} - \min\{Y_i\}) - \frac{1}{2}(\hat{\gamma}_n^X - \hat{\gamma}_n^Y).$$

FIG. 2 shows an illustrative single-server network deployment of present inventive techniques. Specifically, FIG. 2 shows a time server 200 connected through a network 210 to a plurality of other network nodes 220-*i*, *i* = 1, 2, ..., N. Nodes 220-*i* may be routers, switches, servers of various kinds, network end points (including terminals, workstations or computers), or any other kind of network node. Each of nodes 220-*i* has a clock and messaging facilities for exchanging messages with time server 200 in the manner described above. That is, time server 200 forms one of the pair of nodes and, in turn, one or more (typically all) of the nodes 220-*i* forms the other of the node pair for purposes of exchanging time-stamped messages and deriving offset estimates and estimate bias information. While each of the nodes 220-*i* may have equal access to time server 200, priorities may be accorded some nodes 220-*i*, or some nodes 220-*i* may be accorded access to server 200 more frequently.

By exchanging messages with nodes 220-i, time server 200 will provide clock offset estimates and estimate bias information as described above, which information is available at nodes 220-i for correcting clock offset. Of course, N may have a value of 1, so that only a single network node device may interact with a particular time server. While time server 200 is shown as a separate dedicated function network node, it will be understood that the function of network node 200 may be included in a node performing other functions. Likewise, many network arrangements will have a plurality of time servers, each serving network nodes connected on a respective network or sub-network 210.

FIG. 3 shows an illustrative alternative network arrangement in which a plurality of time servers 330 and 340-*i*, *i* = 1, 2, ..., M, are connected in hierarchical relation through a plurality of networks 310-*i*, *i* = 1, 2, ..., M. In the illustrative arrangement of FIG. 3, only two levels are shown in the server hierarchy, but those skilled in the art will recognize that any number of levels of time servers may be used. Likewise, while the number of networks is shown equal to the number of nodes at the lowest hierarchical level, no such limitation is required in practicing the present invention using a hierarchical arrangement of time servers. Each of the networks 310-*i* has one or more network nodes capable of accessing the respective time server connected to the network. By way of illustration, network 310-1 has nodes 350-11 through 350-1P connected to it. Likewise, network 310-M is shown having nodes 350-M1 through 350-MQ. Here, P and Q may be any integer.

In operation, time server 330 exchanges time-stamped messages with each of the time servers 340-i to provide the latter with offset estimates and estimate bias information of the type illustrated above to permit clock correction at the illustrative (second-level) time servers 340-i. Each of the time servers 340-i then serves the clock correction requirements of respective nodes 350-xx in the same manner. Of course, when more than two hierarchical levels of time servers are used, each level (after the first or highest) derives clock synchronization information from a time server at the next highest level. The number of nodes will generally vary from one network 310-i to another, and all or some of networks 310-i may be sub-networks of a larger network. Some time servers may be connected to nodes such as 350-xx and to a next lower order node as well. Some or all time servers may be located in the same local area or distributed over a wide area (including globally) to meet load and geographic distribution requirements for clock synchronization service.

Access to respective time servers by particular nodes (or subordinate time servers) may be scheduled (*e.g.*, periodic), dependent upon availability of time server resources, dependent on prior clock offset behavior at particular nodes (or subsidiary time servers) or detected conditions at such nodes or subsidiary time servers. Exchange of messages and derivation of correction information in accordance with present inventive teachings may be initiated, in appropriate cases, by a particular time server or by a node (or